Automation of Design Flow for Efficient Implementation of Latch Repeater Functional Blocks in Custom Design Approach

Major Project Report

Submitted in partial fulfillment of the requirements
for the degree of

Master of Technology
in
Electronics & Communication Engineering
(VLSI Design)

By

Hardik Jirawala
(13MECV07)

Electronics & Communication Engineering Branch
Electrical Engineering Department
Institute of Technology
Nirma University
Ahmedabad-382 481
May 2015

Automation of Design Flow for Efficient Implementation of Latch Repeater Functional Blocks in Custom Design Approach

Major Project Report

Submitted in partial fulfillment of the requirements
for the degree of

Master of Technology
in
Electronics & Communication Engineering
(VLSI Design)

By

Hardik Jirawala
(13MECV07)

Under the guidance of

External Project Guide:

Ajit N Oke
Component Design Engineer,
Intel India Technology Pvt. Ltd.,
Bangalore.

Internal Project Guide:

Dr. N. M. Devashrayee
PG Co-Ordinator (VLSI Design),
Institute of Technology,
Nirma University, Ahmedabad.

Electronics & Communication Engineering Branch
Electrical Engineering Department
Institute of Technology
Nirma University
Ahmedabad-382 481
May 2015

Certificate

This is to certify that the Major Project entitled **"Automation of Design Flow for Efficient Implementation of Latch Repeater Functional Blocks in Custom Design Approach"** submitted by **Hardik M Jirawala (13MECV07)**, towards the partial fulfillment of the requirements for the degree of Master of Technology in VLSI Design, Nirma University, Ahmedabad is the record of work carried out by him under our supervision and guidance. In our opinion, the submitted work has reached a level required for being accepted for examination.The results embodied in this major project, to the best of our knowledge,haven't been submitted to any other university or institution for award of any degree or diploma.

Dr. N. M. Devashrayee	**Dr. N. M. Devashrayee**
Internal Guide	PG Coordinator (VLSI Design)
Dr. P. N.Tekwani	**Dr. Ketan Kotecha**
Head of Department, EE	Director, IT-NU

Date:

Place:Ahmedabad

Declaration

This is to certify that

1. The thesis comprises my original work towards the degree of Master of Technology in VLSI Design at Nirma University and has not been submitted elsewhere for a degree.

2. Due acknowledgment has been made in the text to all other material used.

- Hardik Jirawala
13MECV07

Disclaimer

Acknowledgements

I would like to express my gratitude and sincere thanks to **Dr.P.N.Tekwani**, Head of Electrical Engineering Department, and **Dr. N. M. Devashrayee**, PG Coordinator of M.Tech VLSI Design program for allowing me to undertake this thesis work and for his guidelines during the review process.

I am deeply indebted to my thesis supervisor **Dr. N. M. Devashrayee**, Professor, E.C.Dept., Nirma University and **Mr.Ajit N Oke**, Design Engineer at Intel India Technology Pvt. Ltd. for their constant guidance and motivation. I also wish to thank **Mr.Saurabh Sharma**, Manager, Intel India Technology Pvt. Ltd., **Mr. Kr, Sanjeev** and all other team members at Intel for their constant help and support. Without their experience and insights, it would have been very difficult to do quality work. Special Thanks to my fellow intern **Mr. Shovan Bhattacharya** for helping me throughout this period.

I wish to thank my friends of my class for their delightful company which kept me in good humor throughout the journey. Last, but not the least, no words are enough to acknowledge constant support and sacrifices of my family members because of whom I am able to complete the degree program successfully.

- Hardik Jirawala
13MECV07

Contents

List of Figures

Abstract

Modern ASIC designs having ultra-deep submicron processes, have interconnects which are very long. RC delay on such interconnects are so large that data on these paths require multiple clock cycles to reach destination. Logically we ave sufficient sequential stages on such interconnect paths, so that it's only matter of splitting the RC delay between sequential stages to solve them. Retiming is the methodology used to split the interconnect delay between one of more sequential stages by adding or changing placement of sequential elements. This retiming can solve the paths with more than one cycle of interconnect delay between them. The sequential element is referred to as a “latch repeater”.

Our goal is to place group of these latch repeaters in one functional block such that they are aligned to their input and output nets, results in optimal interconnect and latch/FF implementation. In the existing mode of work, the input and output interconnects may be routed independently, resulting into possibility of misalignment with latch repeater location, and hence sub-optimal design.

The work details a flow, in which input and output net pairs are found by doing synthesis of RTL code and using scripts to parse logic cone by which we find input output relationship across the sequential. Next step is to obtain open length between this two nets and reduce it to a value as small as possible. Open length is defined as total of misalignment in horizontal and vertical direction between end points of given two nets. Data obtained from this two steps is used to implement the functional blocks which contains sequential.

Chapter 1

Introduction

1.1 VLSI Design Flow

With the technological advancements in field of VLSI chip designing, the complexity of chip designing flow is increased due to high performance required with less power consumption. Continuous shift towards design in nanometer scale has been increasing complexity in physical designing of chip which is one of the most important designing stage. Majority of designing time is consumed by the physical designing stage.

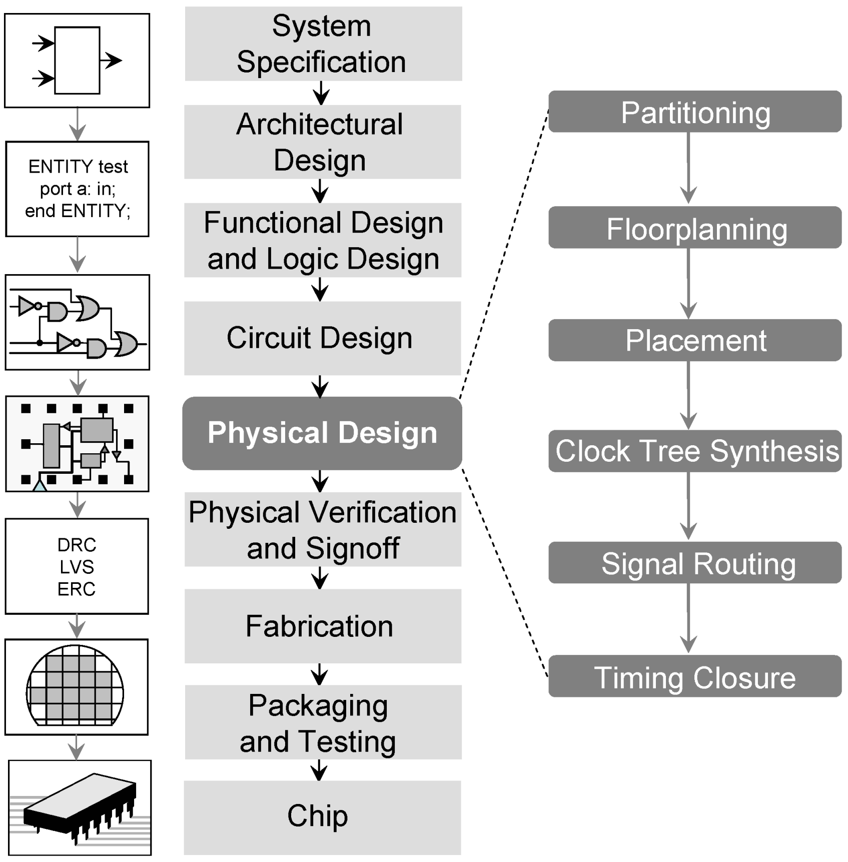


Figure 1.1: VLSI Design Flow

The VLSI design flow shown in 1.1 starts with the specifications and requirements of the market. Behavioral description is generated to analyze the design in different terms like functionality, operating frequency, performance, power, process etc. RTL description is done using different HDLs like verilog or VHDL. This RTLs are then verified for the intended functionality. this RTL are then converted into gate level netlist using logic synthesizers. This netlist are description of gate level connectivity of signals. this gate level netlist is then converted into the Physical design to meet several design aspects like timing, power, area etc.

Physical design converts circuit description into device description. The Physical design cycle consists of Partitioning, Floor planning, Placement, Clock tree synthesis, routing and timing closure.

Partitioning : Dividing a big complex design into number of sub blocks that helps structural implementation by divide and conquer approach. Hierarchical partitioning methodology can localize the engineering changes and decrease the complexity.

Floor Planning : Arranging different structural blocks according to the architecture such that early estimation of chip area, congestion of routing and delays of signals can be done preciously.

Placement : This is the critical step in VLSI design flow because this decides performance, routing availability , heat distribution and power consumption of the full chip. A good placement can control all of this measures of design.

Clock tree synthesis : Clock balancing is important for meeting all timing requirements of the design, mainly in multi-clock designs. this is done after the placement is done. clock routing is given more priority than the signal routing so that clock nets are given good routing resources than the signal routing. Clock spines are routed according to the floor plan of design and the topology of design.

Routing : Main objective of roting is to provide connectivity between all pins. Things to be controlled while routing is to reduce the connecting wire length and making sure that all the timing requirements are met.

Constant shift towards design in nanometer scale has been growing the gap between device delays and wire delays, specifically the global interconnect delays, because they do not scale well with the scaling of transistor size. Traditional methods will not be adequate to meet the performance requirements for high frequency global interconnects. Even with the better-quality metallization and interconnect optimization; the delays on wires that are spread over the chip will exceed the clock period. Subsequently, in modern nanometer designs it is unmanageable to carry signal across the chip in a single clock cycle, which in turn limits the maximum operating frequency of whole design. Different Solutions like new interconnect Material innovations, enhanced design techniques, and unconventional methodologies are necessary to find suitable solution besides low dielectric constant copper metallization and traditional buffer-insertion based interconnect systems. It was predicted that the continued scaling of design geometries, higher frequencies, and more complex chips will make the inconsistency between interconnect needs and projected interconnect performance worse in future.

1.2 Interconnect Pipelining

A concept known as interconnect pipelining, where insertion of sequential elements in interconnects lines; is one feasible solution for modern nanometer technology designs. The idea is to divide a wire whose delay is longer than one clock cycle, by redistributing one or more sequential elements. Two types of sequential elements can be used to create interconnect pipelining 1. Flip flop based pipelining and 2. Latch based pipelining.

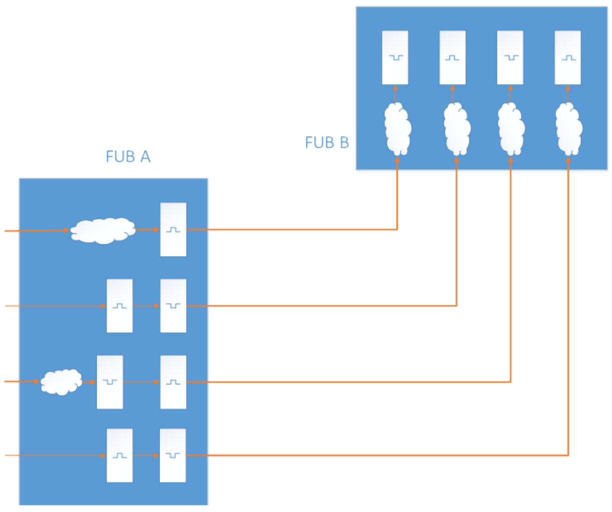


Figure 1.2: Functional Blocks with long interconnect

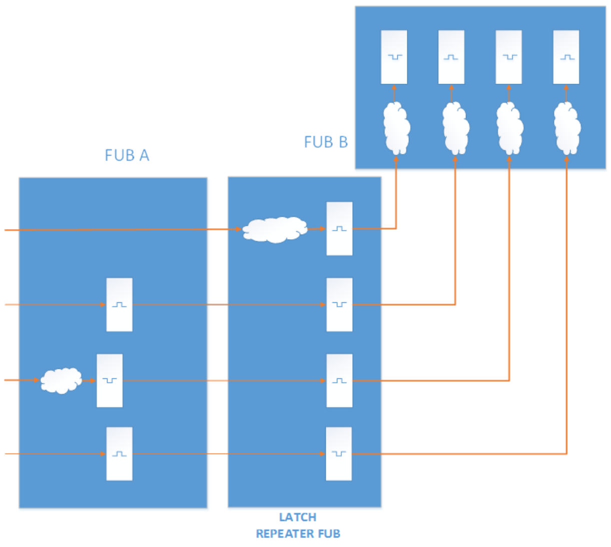


Figure 1.3: Retiming of interconnect paths

1.3 Challenges with interconnect pipelining:

1. CAD tools related issues: Whenever we add a sequential repeater i.e. a flip flop or a latch into a design, RTL changes are also done. All the changes made for repeater insertion are totally manual. Some times when design becomes more stable we need to remove few repeaters also. All this changes in design are done by changing RTL of functional blocks. As RTL changes all the test vectors applied to previous design are no longer useful. So whole design has to be verified by CAD tools again.

2. Architecture related issues: Interconnect pipelining will alter the functionality and clock cycle behavior of design if we insert extra flip flops or latches in the design. To resolve such issue, concept of retiming is used instead of adding extra sequential elements in the design. In retiming concept, some of the sequential elements are moved from the original place to overcome the timing requirement of the Interconnects.

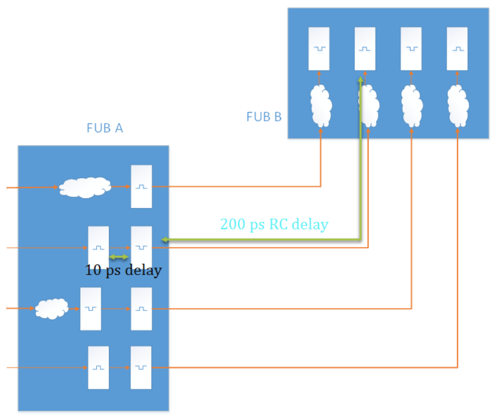
1.4 Advantage of using Latches over flip flops in Interconnect pipelining:

In comparison to flip-flop based wire pipelining, latch-based wire pipelining will acquire a smaller propagation delay, more timing flexibility and will save power and area. The benefits of using level sensitive latches in wire pipelining are particularly the timing flexibility that allows cycle stealing.

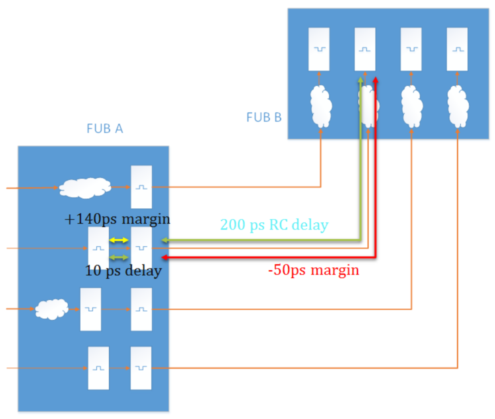
In Flip flop based pipelining scheme there are few constrains. In flip flop based pipelining signal departure time is fixed at either rising or falling edge of clock, whereas in latch based pipelining the signal departure time can be anything in the active period of the clock edge.Also the signal arrival time in flip flop based pipelining is dependent only on wire delay between two flip flops, whereas in latch based pipelining signal arrival time depends on the signal arrival time on previous latch and the wire delay. Due to such constrains, delay between two flip flop cannot be greater than one clock cycle period. But this is not true for latch based pipelining because we can use cycle stealing to compensate the timing between two latches.

One more significant advantage is area consumption. Flip flop consumes almost double area than one latch e.g. one D-flip flop is made of two D-latches. This reduces overall chip area of the design thus reduces power.

1.5 Interconnect Pipelining Example



(a) Blocks with interconnect RC delay before retiming



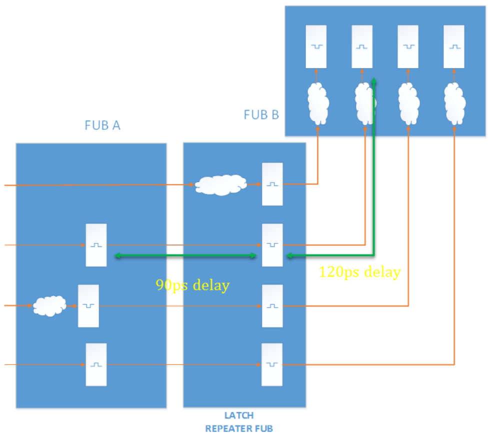
(b) Blocks with margins before retiming

Figure 1.4: Before retiming

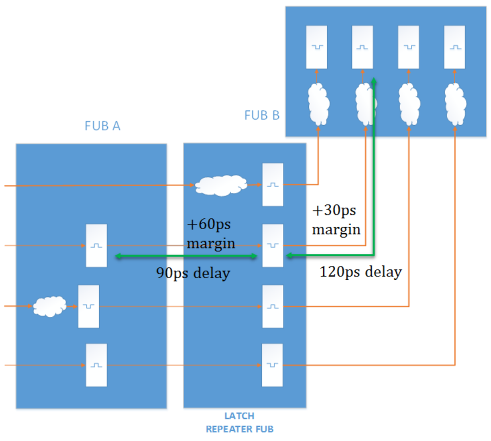
Interconnect pipelining also helps in resolving path having negative margins in the functional block. one example is given here to illustrate how retiming can help resolving a path having negative setup margin due to long interconnect RC delay. assumptions made here are all latches have zero setup and hold time, clock skew is considered as zero and clock periode is 150ps.

Here in figure two functional blocks are shown with interconnect delays and margin. As there is a speed path between two latches in functional block A, there is a positive setup margin between this two latches. due to long RC delay between block A's second latch and block B's first latch, there is a negative margin on last latch on the path. here we can apply retiming on the latches which are between two latches and redistribute RC delay with both the latches connected to this latches.

After retiming we can see here that on latches in block B are connected to latches of latch repeater functional block with interconnect having RC delay less than the clock period. Here we have distributed long RC delay in two parts and thus the setup margin on both



(a) Blocks with interconnect RC delay after retiming



(b) Blocks with margins after retiming

Figure 1.5: After retiming

the latches is made positive. same example can be used to show a solution for a negative hold margin path on the second latch in block A.

Chapter 2

Problem Statement

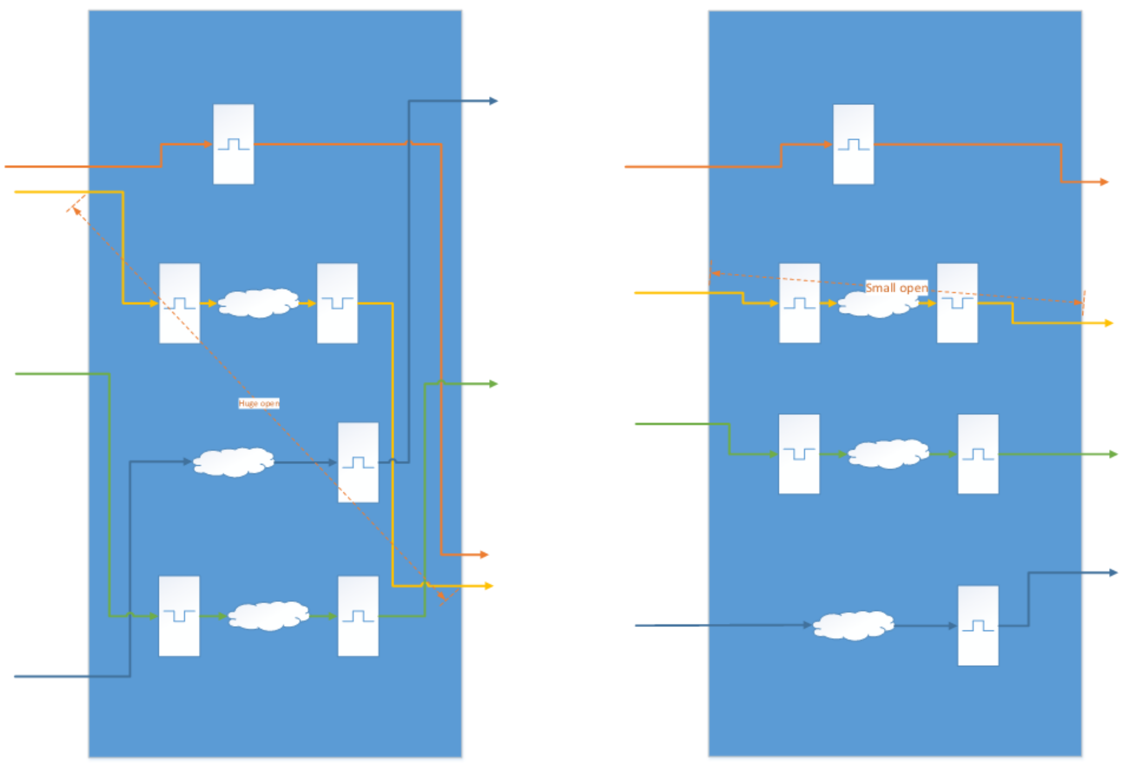
2.1 Problem Defination

At the beginning of a project these latch repeater Functional Blocks are not converged in comparison to other Functional Blocks, as these Functional Blocks have very few amount of logic density than other conventional Functional Blocks. As project progresses there are many design changes being done at circuit level, architecture level, and layout level; which includes RTL changes of all Functional Blocks. As per the Functional Blocks requirement, many of latches may be moved into the latch repeater Functional Blocks and reverse is also possible. Over the time of period, when all this logical blocks start getting implemented and converged to their timing requirements with stable RTL; these latch repeater Functional Blocks become more important to achieve required timing specification interface of Functional Blocks connected to these latch repeater Functional Blocks. So when time comes to implement the latch repeaters, the section routing might be very misaligned over this Functional Blocks. Which defeats the purpose of creating this Functional Blocks, which is to reduce interconnect delay between logical Functional Blocks. So at the end period of project, it becomes very difficult to meet timing on these Functional Blocks in very short period of time. So problem is to obtain the initial implementation of Latch repeater Functional Blocks done at very early stage so that Design Engineers' time can be saved.

2.2 Proposed Solution

Proposed solution to overcome this issue is to maintain good alignment of the section routing from very beginning of the project timeline till the end. But in present mode of work, the input and output nets of this Functional Blocks are routed independently so there are very high possibility of misalignment of these nets.

But question here is, which nets need to be aligned. So one of the major concern is to



(a) Before alignment (b) After alignment

Figure 2.1: Latch Repeater Before and After Alignment

get such nets which are connected through a sequential elements. So here are the steps to get proper alignment of the nets.

1. To obtain pairs of pins of Functional Block which are logically connected through a Latch or a Flip Flop.
2. To get the net connected to this pins.
3. To obtain open distance between the end points of the nets.
4. Move nets closer to each other so that the open distance become very small.

Once we get proper alignment then using the net location and net pair connectivity we will be able to implement the whole latch repeater Functional Block and give it to designers as an initial implementation of design.

Here in Figure 2.1a one latch repeater with misaligned input-output pair, latches and with some logic cloud is shown. Also a possible solution to the misaligned nets is also shown in 2.1b.

2.3 Why ICC based P&R tool is not used?

The main reason why ICC based synthesis and placement flow is not used for such functional blocks, is clock tuning difficulties and need for careful sizing. These functional blocks are very tall or wide, and automated tools do not perform optimally for their implementation. These blocks, therefore, are designed using the structured data path design styles, and are costly in effort.

This new work flow details, a simple DC synthesis utility to extract pairs of input and output pins from the RTL code. These pairs of pins are linked to the section netlist and passed to section routing tool. Open length between the pair of nets over the surface of the repeater functional blocks is found in section layout using a utility developed specifically for this task. A suitable reporting structure and additional set of utilities are created to help the section layout owner to correct and verify the routing for the pairs of nets.

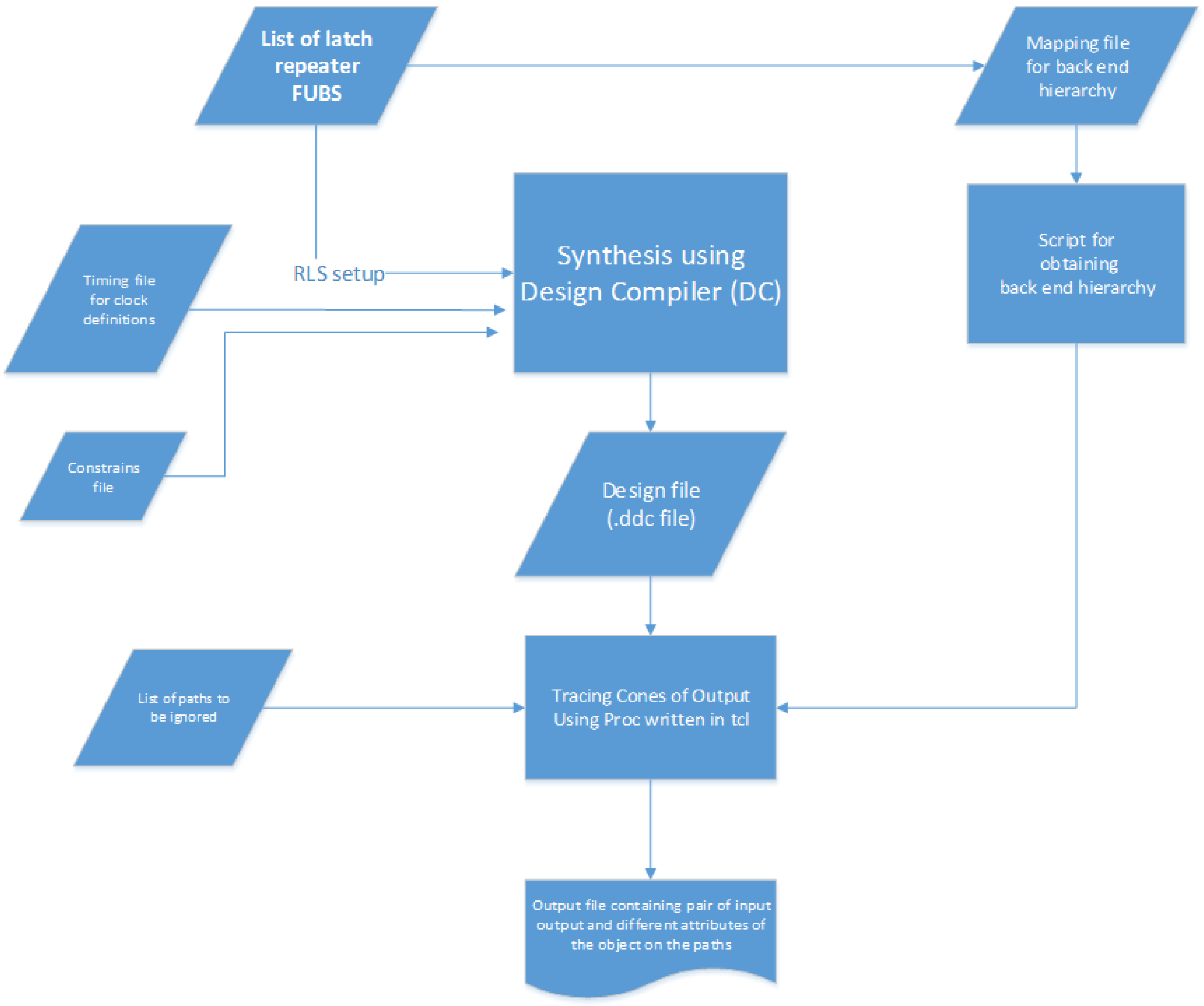
Chapter 3

Proposed algorithm

First task is to get the logical connectivity between input net and output net. to accomplish this task we have created an algorithm which can be applied to a design compiled with Design Compiler®

A flow chart for getting this data file is shown in figure 3.1 . for each latch repeater Functional Block we have to follow certain steps to generate connectivity file which is used for next steps.

1. Setup an environment for to do synthesis process using DC particular Functional Block.
2. Copy all the collateral files e.g timing information file, synthesis flow control file, layout information file etc. to their respective Destinations.
3. Get the Back-end Hierarchy of the Functional Block so we can know where to look for nets in Layout.
4. Synthesize the Design from the RTL stored in central datbase.
5. Apply Trace back algorithm script on the design file (.ddc file) generated after the synthesis. this algorithm will generate a text file which contains information about connectivity between input and output nets and different cells & their properties. This flow also generates a netlist file (.sch file) which will be used to implement the Functional Block.



List of latch repeater FUBS
Mapping file for back end hierarchy
Timing file for clock definitions
RLS setup
Synthesis using Design Compiler (DC)
Script for obtaining back end hierarchy
Constrains file
Design file (.ddc file)
List of paths to be ignored
Tracing Cones of Output Using Proc written in tcl
Output file containing pair of input output and different attributes of the object on the paths

Figure 3.1: Flow Chart to get Input Output Connectivity

3.1 Trace Back Algorithm

A basic circuit diagram for trace back algorithm is shown in figure 3.2 . Steps to trace back path are listed below.

1. List down all the output ports in on list.Foreach output from this list we will perform following steps.

2. We will get the startpoints of the timing paths ending to that output. For any port or a pin, startpoints are the input nets of the first sequential element that comes on a trace back path or an Input pin which has a logical path directly to the output port. For Example in figure 3.2 Output X has startpoints as Input A, Input E and input pin of Latch A. At this step we store the latch property like type of latch and the size of a latch in one list.

3. If startpoint is an input port then report it to the output file, as pair of input and output pin.

4. If startpoint is not input in then for that startpoint repeat step 2.

5. Repeat steps 2,3 and 4 until we get all the input pins having connectivity to the output pin through a sequential element or a logic cloud.

6. Repeat all previous step for all the Output pins.

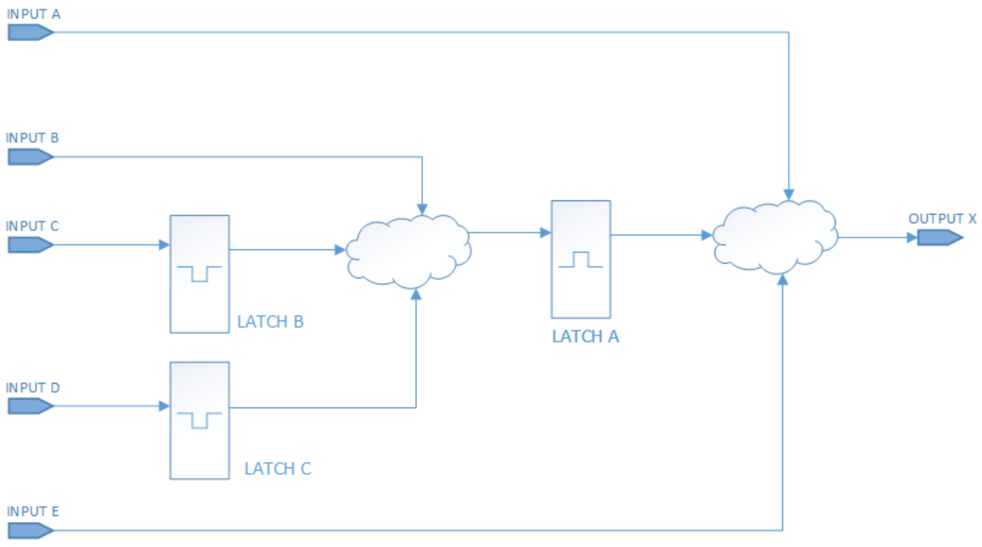


Figure 3.2: Basic Diagram for Algoritm to get Input Output Connectivity

3.2 Algorithm to find the open length between two nets

Here are the steps to get the open Length between two nets.

1. From output file of Previous stage we get a pair of pins and hierarchy of the Functional Block over which we should align the nets.

2. Next step is to get the name and geometry of the nets connected to those pins on full chip layout.

3. Once we get geometry of nets, we find out if net has a via on it over the Functional Block or not. we also get the child nets of the parent net.

4. From all this possible points we calculate the smallest open distance between two nets and report it in the output table.

5. From geometry of the Functional Block we get the data if net is routed over particular Functional Block. if net is not routed over the Functional Block it will indicate very big open distance between those two nets in output table.

6. Repeat steps all other steps for each pair of previous flows output file.

7. Once we get all the open lengths for each pair, we sort them in Descending order.

After this step, we give output data to SLO's (section layout owners) so that they can look at the net pairs having large opens and align these nets to reduce open length. Here in Figure 3.3 one bus having 64 bits was misaligned over one of the Functional Block. Using over scripts we got the logical connection between each pair and tried to align those nets in one track. Hence after alignment was done the scenario was looking as Figure 3.4. Here seeing criticality of these signals, it become very useful for designers to meet proper timing requirements with minimum RC delay in section layout. This algorithm also generates a text file which contains information about (x,y) co-ordinates of input and output net's endpoints. This information will be used in the next step to place cells connected to those nets.

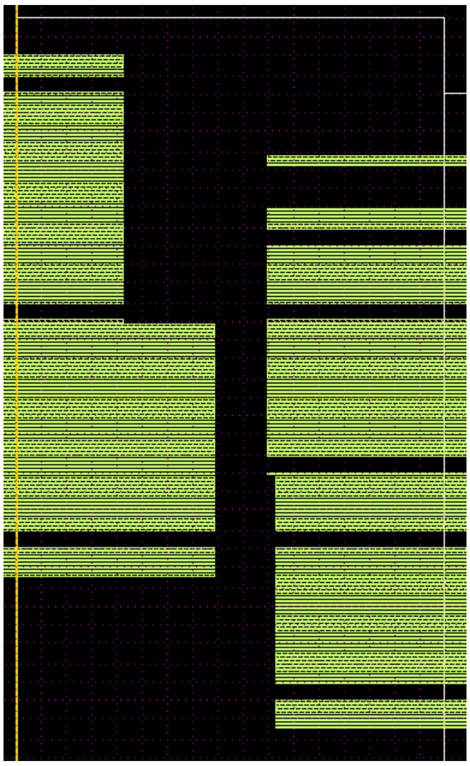


Figure 3.3: Misalignment between input nets and output nets

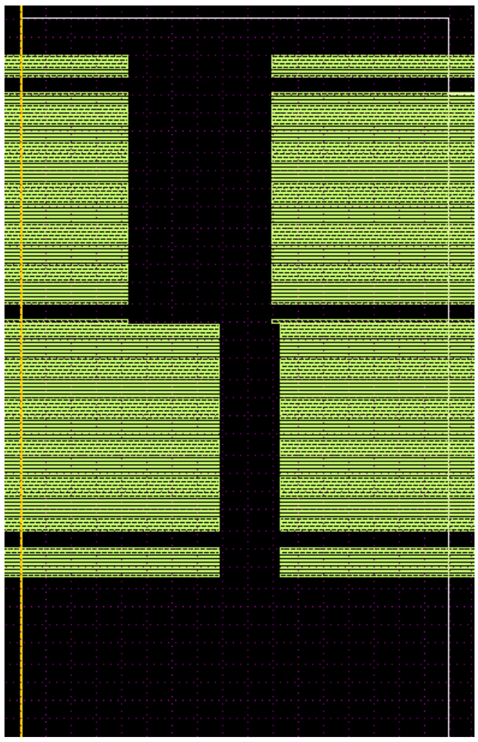


Figure 3.4: Alignment done using our flow

3.3 Algorithm to Place Cells According To Section Routing

Once SLO's have reduced open length between input and output nets, we can get new (x,y) co-ordinates of this nets and use this information to implement Latch repeater functional blocks.

Steps to implement cells according to section routing.

1. Get the highest and lowest values of horizontal and vertical co-ordinates , to decide the boundary of the functional block.
2. Take one pair of input and output net and create a rectangular block having diagonal points as end points of nets. Also get the center of this block.
3. Get cells connected to input & output nets from schematic file and place them on the centre of rectangular block.
4. Place all cells for all pairs having routing over the functional blocks.
5. Place all cells which don't have section routing over the functional block to the Origin of functional block.
6. Remove all the overlaps of the cells and get a clean placement of the design.

This step will generate a initial design that can be given to design engineers to do some incremental changes to meet different design challenges like timing, area, power, reliability etc. As this design is flat design (without any hierarchy), we can also use some optimization tools which works very good on the flat design. Some of these tools are good at reducing power and area requirement of the cells. This even further reduces efforts for designer to converge the design in short time.

3.4 Challenges and limitations

1. Initially the RTL synthesis had the capability to reach till 3 or 4 depths of logical connectivity, but now we are able overcome this limitation and flow is capable of going till any depth of logic connectivity. Thus we can get the whole logical cone of an output for any complex design.

2. algorithm was level dependent at first i.e. it was not showing satisfactory results when we move deeper into the block hierarchy.

3. In the placement flow, cleaning Overlaps was a tough task. automation can not completely remove overlaps of cells. Some efforts are required by designer to overcome this situation.

Chapter 4

Conclusion

In this study, a work flow for implementation of latch repeater functional blocks according to aligned input and output section routing is presented. Using this flow, section routing for input net and output net was aligned for all the latch repeater functional blocks. This helped in doing fast integration over this functional blocks with saving lot of higher metal resources in the section. Implementation of four latch repeater was done using this flow. According to design engineers working on this latch repeater functional blocks, the placement of cell was 60-70% similar to what designer would have done.

Bibliography

[1] Jingye Xu and Masud H. Chowdhury et al., **"Latch Based Interconnect Pipelining For High Speed Integrated Circuits"** *E*lectro/information Technology, 2006 IEEE International Conference, San East Lansing, MI, 7-10 May 2006.

[2] Seth, V; Min Zhao ; Jiang Hu et al., **"Exploiting Level Sensitive Latches in Wire Pipelining"** *C*omputer Aided Design, 2004. ICCAD-2004. IEEE/ACM International Conference, 7-11 Nov. 2004

[3] http://www.intelpedia.intel.com

[4] http://www.dts.intel.com

Acronyms

SLO	Section Layout Owner
VLSI	Very Large Scale Integration
RTL	Register Transfer Level
DC	Design Compiler
IC	Integrated Circuit
ICC	IC Compiler
CAD	Computer Aided Design
HDL	Hardware Description Language
VHDL	Very High Speed Integrated Circuit HDL